Exercice Pratique - CM1 part 2

Les exercices suivants sont composés en deux parties distinctes:

I. **Manipulations basiques d'une database**
II. **Création de requêtes SQL**

I. Manipulations basiques d'une database

1. Créer les tables: orders, products, merchants, et users avec les variables suivantes:

Orders	Products	Merchants	Users
-id int [pk] - user_id int [not null, unique] - status varchar (orders_status) enum - created_at varchar	-id int [pk] -name varchar -merchant_id int [not null] (ref Merchants) - price int - status varchar - created_at varchar	- id int [pk] -created_at varchar	-id int [pk] -full_name varchar -email varchar [unique] -gender varchar -date_of_birth date -created_at date -country varchar

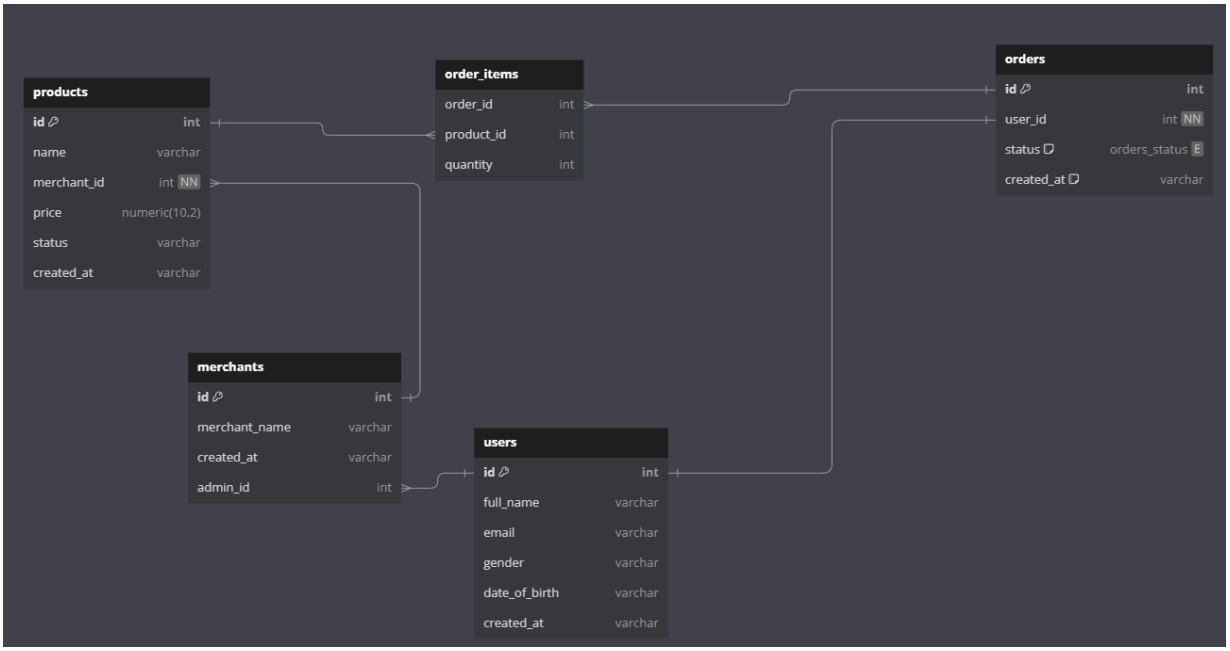
Sachant que order_status contient les valeurs suivantes:

- created
- running
- done
- failure

Il faudrait aussi créer une table order_items qui permet de lier les tables Orders et Products (many-to-many)

2. Pour les modifications suivantes, évitez de supprimer la table pour la recréer… :
    a. Il faudrait rajouter une colonne merchant_name en varchar pour la table Merchants, et une colonne admin_id qui est une Foreign Key issu de la table Users
    b. Supprimer la colonne country de la table Users
    c. Modifier le type de la colonne price de Products pour la faire passer de int à numéric(10,2)

3. À la fin, vous devriez avoir le Data Model suivant:



4. Pour la table Users, ajoutez les valeurs suivantes:

id	full_name	email	gender	date_of_birth	created_at
1	Eric Dupont		F	01/01/1807	06/12/2023
2	John Smith	lebest@gmail.com	M	23/08/2000	02/10/2003
3	Sylvain Durif	grandmornarquecosmique@hotmail.fr	M	25/12/2099	01/01/1205

5. Supprimez toutes les tables. Elles ne vont pas être utiles pour la suite.

Exercice Pratique - CM2

II. Création de requêtes SQL

1. Pour pouvoir s'entraîner sur une database déjà construite, importer sur PostgreSQL la database de :

https://github.com/vrajmohan/pgsql-sample-data

2. Afficher la table **employee**

3. Calculer le nombre d'employés

4. Donner la liste des employés dont le nom de famille commence par A. dont le nom de famille fini par H, dont le nom de famille à un h en 3ème lettre

5. Afficher les 20 plus jeunes employés

6. Calculer le nombre de dates de naissances différentes

7. Donner la liste des employés:
    a. Arrivés après 1997
    b. Arrivés après Juillet 1997
    c. Arrivés entre 1994 et 1998
    d. Arrivés entre Novembre 1994 et Septembre 1995

8. Trouver l'employé le plus payé, puis le moins payé(donner le nom, prénom, salaire annuel)

9. Calculer la somme que l'entreprise à verser pour payé ses employés en Février 2000 (On va supposer que la paye est versé tout les 25 du mois, que les payes suite à une augmentation se font en mois complet, ...)

10. Donner le nombre d'employés par département et par ordre décroissant le 15 Janvier 2000 (Afficher aussi id du department ainsi que son Nom)

11. Trouver les id, nom, prénom des employés, qui ont changés de department dans la table department_employee

12. Quel department est en moyenne le plus payé actuellement au centime près? (en supposant que personne n'est parti depuis les années 2000)

13. Créer une nouvelle colonne dans employee is_senior_engineer pour voir si l'employé à le titre de 'Senior Engineer" ou non

14. Mettons-nous dans le cas où on voudrait supprimer des données de la table employee. Quelle serait la requête pour supprimer les gens arrivés avant le 15 Février 1990 (ne faite pas tourner).

III. Challenges

1. Trouver les employés qui gagnent plus que leurs Managers

2. Trouver l'employé qui a le salaire le plus élevé par department

3. Trouver les employées qui ont l'un des 3 meilleur salaire dans chaque département

4. Afficher le nombre de personnes recrutées par an/mois (à voir), puis l'augmentation/diminution d'employé recruté par an/mois